COMPUTER SCIENCE COMPETITION - JAVA TOPIC LIST 2026-2027

IMPORTANT NOTES: Java is the official programming language for UIL Computer Science. Contest content for 2026-2027 will conform to the Java Platform Standard Edition, Version 26. This list is intended as a guideline and is not all-inclusive. Knowledge of basic Java concepts is assumed. Visit the UIL web site at uiltexas.org/academics/stem/computer-science for a list of Java resources and other important contest information.

Primitive types (`int, double, boolean, short, long, byte, char, float`), casting of primitives
Arithmetic operators (`+, -, *, /, %, ++, --`) and string concatenation
Assignment operators (`=, +=, -=, *=, /=, %=`)
Increment and decrement operators (`++, --`) and their values in expressions
Boolean expressions and operators (`==, !=, <, <=, >, >=, &&, ||, !, &, |, ^`) including short-circuit evaluation
Bitwise operators (`<<, >>, &, ~, |, ^, >>=, <<=, !=, &=, |=, ^=`)
Branching (`if, if/else, ?:, switch, break`)
Looping (`while, for,` enhanced `for, return, do/while, break, continue`)
`System.out.print(),println(),printf(), %f, %d, %s, escape sequences \n, \\, \"`
Parsing (`String.split(), Integer.parseInt(), Double.parseDouble()`)
User-Defined Classes (constructors, methods, instance variables, private vs. public, overloading, overriding, `final` local variables, `static final` class variables, `static` methods, `static` non-`final` variables)
Object-Oriented Principles (inheritance, abstract classes, encapsulation, interfaces and polymorphism)
Conversion to supertypes and (`Subtype`) casts
Comparison of reference types (`equals(), ==, !=, Comparable.compareTo()`)
Enumerated Data Types
Arrays, including arrays of arrays and initialization of named arrays
Generic collections (`Collection, List, Set, Map, Stack, Queue, PriorityQueue, ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap`) – *See supplemental class reference list.*
Regular expressions using Pattern and Matcher
`Arrays.sort()` and `Collections.sort()`
Java Standard Library (`String, Integer, Double, Character, Math, Object, Comparable, Scanner, File, Random, Arrays, Assert`) – *See supplemental class reference list.*
Lambda Expressions and Functional Interfaces *(see supplemental guide on UIL CS website)*
Data Structures: Arrays and Lists, Stacks and Queues, Linked Lists, Trees (BST, AVL Tree, Red-Black Tree, etc.), Heaps and Priority Queues, Hash Tables, Graphs, Tries, Disjoint Set Union, Segment Trees, Fenwick Trees
Sorts (Selection, Insertion, Mergesort, Quicksort, Radix, Bubble) and Searches (Sequential, Binary) – *same canonicals as AP*
Recursion, Dynamic Programming *(dynamic programming applies only in the programming portion of the contest, not the written tests)*
Greedy Algorithms (Activity selection, Fractional knapsack, Huffman coding tree, etc.)
Graph Algorithms (BFS, DFS, Dijkstra's, Kruskal's, Prim's, Topological sorting, cycle detection, etc.)
Computational Geometry (Convex hull, line intersection detection, closest pair of points, etc.)
Probabilistic Algorithms (Monte Carlo PI estimate, randomized quicksort, reservoir sampling Floyd's cycle detection, etc.)
Analysis of algorithms: informal comparison of running times, exact calculation of statement execution counts, Big-O notation, best case / worst case / average case time and space analysis
Base Conversions and Arithmetic
Two's complement binary representation of negative 8-bit integers – conversion both ways between base 10 and base 2
Boolean simplification using generic notation (A*B, A+B, A⊕B, $\overline{A}$, $\overline{A * B}$, $\overline{A + B}$, $\overline{A \oplus B}$ – using truth tables and Boolean Identities to analyze and simplify Boolean expressions *(see list of identities on UIL CS website)*
Polish notation – representation, analysis, and conversion of simple infix, prefix, and postfix expressions
Finite State Machines
Languages and Grammars (formal grammars, BNF, language classification)
Design Patterns (e.g. Singleton, Factory, Strategy)
Multithreading and Concurrency Basics
Functional Programming Features in Java
Digital Electronics – symbolic representation of Boolean expressions using logic gates NOT, AND, XOR, OR, NAND, NOR, NXOR

UIL Computer Science Topic List 2026-2027

Written Test First 15 Questions

1. **Number base concepts**, arithmetic, conversion
2. **Simple literal math expression** with mixed operations
3. **Simple output** involving print, println, printf (%d, %f, %s, \", \\, \n)
4. **String class methods**
5. **Simple Boolean logic** (AND, OR, XOR, NOT) - Java based
6. **Math class methods** (no advanced topics like trig - save that for later in the test)
7. **Simple variable expression** with mixed operations
8. **Conditionals** (if, if/else, switch - not ternary)
9. **Simple output loop**
10. **1D primitive array**, basic concepts
11. **Input concepts –** use of Scanner and File classes
12. **Accumulation loop** – summation, product accumulation, etc.
13. **Order of operations** (beyond just the math expressions - testing knowledge of the full Java spectrum of order of precedence)
14. **Java specific data type concepts**, memory size, max and min limits, wrap around, complements (~)
15. **ArrayList –** generics only

Written Test Last 2 Questions

- The final 2 written test questions (#39 and #40) will be free response with one discrete answer each.

Programming – the first two programs of any packet will be as follows:

- A program that requires only output and will not read data from a file
- A program that will read from a data file, but will be very basic in nature, with little or no significant computation involved

Programming Solutions Runtime

- All solutions should need no more than 10 seconds to run. Those taking longer may be judged as incorrect due to a runtime error.